ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/rajendraprasad-chittimalla/279

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1

Research Article

An Application Integration - SFTP and GCS Connector! Build SFTP to Google Cloud Storage Data Pipelines with Easy-To-Use Data Connectors

Rajendraprasad Chittimalla*

Rajendraprasad Chittimalla, MS in Information System Security, Software Engineer - Team Lead, Equifax Inc, USA

Citation: Chittimalla R. An Application Integration - SFTP and GCS Connector! Build SFTP to Google Cloud Storage Data Pipelines with Easy-To-Use Data Connectors. *J Artif Intell Mach Learn & Data Sci 2022*, 1(1), 1209-1212. DOI: doi.org/10.51219/JAIMLD/rajendraprasad-chittimalla/279

Received: 02 August, 2022; **Accepted:** 18 August, 2022; **Published:** 20 August, 2022

***Corresponding author:** Rajendraprasad Chittimalla, MS in Information System Security, Software Engineer - Team Lead, Equifax Inc, USA, E-mail: rajtecheng4mft@gmail.com

ABSTRACT

This work explores automating secure data transfers between SFTP servers and Google Cloud Storage (GCS) using data connectors. The immutability of GCS objects necessitates frequent transfers for data modifications, creating security vulnerabilities. SFTP to GCS pipelines address this by leveraging secure file transfer protocols. However, traditional approaches involve complex configurations and manual scripting, leading to inefficiencies. Data connectors offer a streamlined solution with pre-built configurations and user-friendly interfaces, reducing development time and human error.

Keywords: SFTP, GCS, data connectors, data pipelines,

1. Introduction

Google Cloud Storage (GCS) is one of the three largest commercially available cloud suites, behind Azure and AWS. GCS is an elemental part of the Google Cloud that is used specifically for storing any form of data, such as immutable objects in containers called buckets. The immutability is one of the features that make CGS different from other cloud storage services, as it prevents any changes from being made to an object once it's in the GCS, and a different version of the file has to be uploaded to make and incorporate any changes. Since GCS focuses on object storage, any edits require downloading the file locally, making the changes, and then uploading the modified version back to GCS. This creates more data transfer instances, and it stretches the attack surface since each transfer instance may lead to new exploitation opportunities. One way to augment security here is to build a Safe File Transfer protocol (SFTP) to GCS data pipelines, which is just one of the ways a secure GCS pipeline/data transfer patch can be created, but developing and creating pipelines for different transfer instances comes with its own set of problems. Therefore, automating the process with the right data connectors can lead to better results.

2. Literature Review

There is ample literature on SFTP as it's a universally used protocol for safe file transfer, covering everything from its inception, implementation, limitations, etc[1]. Adequate literature is available on GCS as well, both from the source (Google) and third-party entities[2]. The concept of data pipeline is discussed extensively in the literature as well. This includes modeling techniques/approaches for pipelines and relevant protocols like ETL (Extract-Transform-Load) and ELT (Extract-Load-Transform)[3]. The literature also discusses frameworks for developing data pipelines for specific use cases like manufacturing[4]. Other areas of focus include discussing the scope of data pipelines, their position in the data lifecycle, and

their optimization in modern implementations and tech domains like Machine Learning (ML)[5].

As core elements in data pipelines and other data transfer use cases, data connectors are an important research topic as well, though the literature is relatively limited and focuses on interpretation, specific connector implementations, use cases, and configurations[6].

3. Problem Statements

An SFTP to GCS pipeline is a solution to the security vulnerabilities inherent in an exposed GCS transfer, though it's not the only one. A few different solutions can be implemented to make GCS transfers even more secure than an SFTP pipeline despite their high frequency (due to the immutability factor). However, that's the most viable implementation for businesses that rely upon SFTP extensively for most of their outgoing and incoming file transfers. But it comes with its own set of challenges/problems.

Inefficiencies in Data Transfer

Traditional data pipeline approaches often involve custom scripting or complex configurations for each or a set of transfer instances. This can lead to time-consuming development, maintenance overhead, and difficulties in scaling data transfer operations. This may drive up a business's cost of operations (costs pertaining to Managed File Transfers or MFTs), increase the chances of data leaks/interception, require more manual oversight, and lead to inconsistencies, even in similar transfer instances. This makes the process of data transfer to GCS inefficient as a whole.

Changing Configuration Requirements

Data transfer instances, even if both source and destination remain the same (local or cloud SFTP servers and GCS), may vary greatly based on different use cases and some other variables. This is something data pipelines need to adapt to and if they are reconfigured for each transfer instance and need to accommodate its complex requirements, it comes with significant time and resource requirements. It significantly reduces the responsiveness and agility of the MFT department/ personnel within a firm.

Vulnerabilities in Misconfigured Pipelines

Complex data pipeline configurations and frequent reconfigurations may enhance the threat/vulnerability profile of SFTP to GCS pipelines, especially when the configuration is done manually.

File size limitations from Frequent Transfers

Frequent transfers are an inherent characteristic of any transfer configuration/arrangement that involves GCS. The reason is the immutable nature of an object on the GCS, which triggers a new transfer to accommodate each change/modification. There is 20 MB size limit for each transfer with GCS Connectors.

4. Solution: Building SFTP to GCS Data Pipelines with Easy-To-Use Data Connectors

Automating transfers between an SFTP server (or cloud) and GCS via data pipelines that use data connectors can solve or at least mitigate the above-mentioned problems.

Efficient Data Transfers

Easy-to-use data connectors can streamline data transfer configurations for SFTP to GCS pipelines. These connectors may offer a graphical user interface (GUI) for ease of use as well as pre-built configurations, eliminating the need for custom scripting and reducing development time. This simplifies data transfer management, lowers maintenance overhead, and facilitates scaling operations by automating routine tasks. As a consequence, both financial and personnel requirements for data transfer tasks can be reduced quite significantly.

Configuration Adaptability

Data connectors can offer flexibility to accommodate diverse data transfer needs. They can be designed to handle various configurations and adapt to different use cases. This eliminates the need for complex (manual) reconfigurations for each transfer instance and allows for easy integration with various data sources and pipeline arrangements between SFTP and GCS. This flexibility empowers the MFT teams to respond quickly to evolving data transfer requirements, enhancing overall agility.

Avoiding Misconfigurations

Data connectors can automate secure data transfer processes, mitigating vulnerabilities arising from manual configurations[7]. They can come pre-configured with robust security features like encryption and authentication protocols, eliminating the risk of human error during configuration. This adds another layer of security over the SFTP to GCS transfers, reduces the attack surface, and strengthens the overall security profile of SFTP to GCS pipelines.

Mitigating the Risk of Frequent Transfers

It's important to understand that even automated data transfer pipelines created using data connectors cannot reduce this risk at its very core, i.e., by reducing the number of transfer instances. However, they can optimize data transfer workflows within SFTP to GCS pipelines and reduce human risk by automating the entire process.

5. Solution Implementation

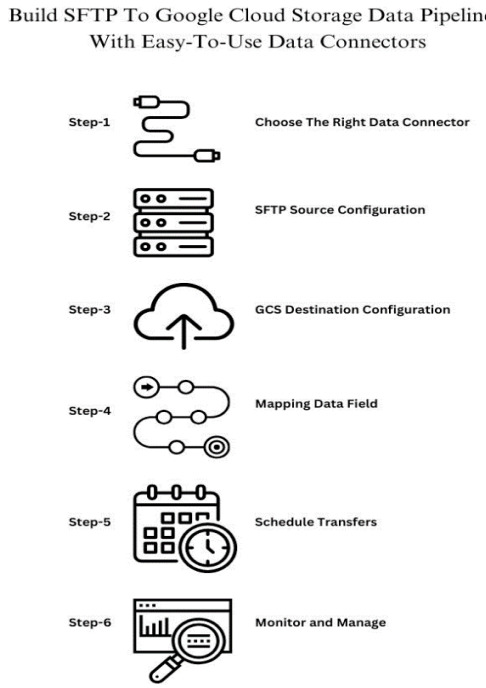

Figure 1: Building SFTP to Google Cloud Storage Data Pipelines with Easy to use Data Connectors.

An SFTP to GCS pipeline is easy enough to automate using

the right connector. The following steps can take you through the process.

Step 1: Choose the Right Data Connector

There are multiple data connector options for SFTP to GCS data pipelines, including an option native to Google Cloud. The Google Cloud Storage Transfer Service allows data transfer (including bulk transfer) between GCS and a variety of other sources, including on-premises servers and other clouds, that may serve as SFTP sources or SFTP access points. It's a managed file transfer service offered by Google and includes multiple layers of security, including encryption during transit. The cost is based on the amount of data you transfer.

Another option is a commercial data integration tool/platform like Talend and Informatica Cloud Data Integration. They are usually well-maintained products, easy to use, and with decent support.

Lastly, there are open source options like Apache NiFi and Airflow[8]. They are maintained by the community and are usually free to use, though the interface may not be as user-friendly as a commercially developed and maintained product.

There are multiple factors to take into account when making this choice, including the cost, user-friendliness of the connector, additional safety features, etc.

Step-2: SFTP Source Configuration

While the exact configuration process may differ from connector to connector, a few general elements common in each SFTP source configuration are host, port, authentication, and source directory.

The host configuration requires using the hostnames and IP address of the SFTP server from where the transfer will be initiated.

Port is about the physical connectivity element of the transfer. It can either be the default port or a custom port, as required by the SFTP server.

Authentication and user credentials for transfer instances can be made part of the configuration for additional security.

The directory setting refers to the exact directory/address within the SFTP server where the transfer will initiate.

Step-3: GCS Destination Configuration

The other end of a transfer, i.e., destination, needs to be configured alongside the source. Again, the process may differ from connector to connector, but there are three core elements you have to take into account: Bucket name, access permissions, and folder structure.

The bucket name specifies the bucket within a GCS project where the file will be stored.

Folder structure becomes relevant if the file has to be stored within a specific folder inside a bucket (the exact destination of the file to be transferred).

Access permission (if set) would ensure that the GCS has relevant credentials to access the file/accept the file from the SFTP source.

Step-4: Mapping Data Field

Once you have already configured source and destination, you have to map the entire data pipeline that requires one additional step: Data transformation. If the data needs to be transformed for any reason between source and destination, it has to be done here.

Step-5: Schedule Transfers

An important step in automating file transfers between an SFTP source and GCS destination is scheduling the transfers to accommodate the usual transfer needs of the organization. Instead of setting an actual schedule that might be a bit rigid, it's possible to take advantage of a few scheduling variables.

- Setting a frequency of transfer. It can be hourly, daily, weekly, etc., based on volume and other transfer-related needs.
- It's possible to set up a time window when the transfer occurs, to ensure they are not coinciding with other outgoing or incoming transfers that may be relying upon the same bandwidth and SFTP computing resources, undermining both.
- The retry mechanism built into the connectors can let you set conditions like how many retries the pipeline should attempt in case of a failure and its timeline.

Step-6: Monitor and Manage

- It's important to monitor and manage data pipelines to ensure they are working as intended.
- Utilize monitoring tools like Google Cloud's Stackdriver to track the status of data transfers. This allows you to proactively identify any potential issues.
- Implement error-handling mechanisms to address any issues promptly. This includes logging errors and notifying administrators.
- Regularly review and optimize the performance of the data pipeline to ensure it meets business requirements.

6. Key Limitations and Important Considerations

Even with pre-configured security features in connectors, it's essential to follow security best practices. This includes strong authentication protocols, access control for GCS buckets, and encryption of sensitive data during transfer.

Reduced Transfer Frequency (Immutable Objects): Data connectors cannot fundamentally change the inherent characteristic of GCS, where objects are immutable. Frequent transfers due to data modifications are unavoidable. However, connectors can optimize transfer workflows.

Vendor Lock-In (Commercial Options): While commercial data connectors offer user-friendly interfaces and features, they can lead to vendor lock-in. Switching to a different connector might require reconfiguring the entire pipeline.

Learning Curve (Open-Source Options): Open-source connectors like Apache Airflow offer flexibility but require technical expertise to set up and maintain. This can be a challenge for teams without experience in data pipeline development.

Complexity for Multiple Transfers: While connectors simplify configurations, managing numerous data pipelines with different configurations can become complex. Data orchestration tools might be used to manage multiple pipelines.

7. Research Impact

A comprehensive understanding of data pipelines, their

configuration and implementation, and researching the best connector for your SFTP to GCS data pipeline needs can help a massive range of businesses that rely on SFTP and have to connect with internal or external GCS servers regularly. It allows businesses to make the process more efficient and safer, mitigate the attack surface, and protect important business and client data in transit while taking advantage of the immutability aspect of GCS. This leads to better productivity and efficient use of MFT resources, enhances the market's/clients' trust in the business's commitment to data integrity, and forestalls legal and reputational liabilities.

8. Conclusion

A data-connector-based pipeline for SFTP to GCS transfer can be transformational for an organization's MFT department. SFTP integrating with GCS Connectors can lead to a successful file transfer solution with being schedule pull from buckets, push to buckets or to a server. It could be a new innovation to SFTP application integration. It mitigates or eliminates several threats to frequent file transfers required due to the immutable nature of objects in GCS buckets and significantly removes the human error element from the equation. However, the success of these data pipelines relies upon choosing the right connector, and continued success warrants monitoring and management.

9. References

1. Arif H, Hajjdiab H, Harbi FA, et al. A Comparison between Google Cloud Service and iCloud. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). Singapore, 2019.
2. Dehury CK, Jakovits P, Srirama SN, et al. TOSCAdata: Modeling data pipeline applications in TOSCA. Journal of Systems and Software, 2022.
3. Firdausy D, Silva PD, Sinderen MJ, et al. Semantic Discovery and Selection of Data Connectors in International Data Spaces. Interoperability for Enterprise Systems and Applications, I-ESA 2022. Valencia, 2022.
4. Nawej CM, Owolawi PA. Evaluation and Modelling of Secured Protocols' Spent Transmission Time. 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC). Mon Tresor, Mauritius, 2018.
5. Oleghe O, Salonitis K. A framework for designing data pipelines for manufacturing systems. Procedia CIRP, 2020.
6. Petrova-Antonova D, Iva Krasteva, SI, et al. Conceptual Architecture of GATE Big Data Platform. CompSysTech '19: Proceedings of the 20th International Conference on Computer Systems and Technologies, 2019.
7. Plale B, Kouper I. Chapter 4 - The Centrality of Data: Data Lifecycle and Data Pipelines. In Data Analytics for Intelligent Transportation Systems, 2017.
8. Raj A, Bosch J, Olsson HH, et al. Modeling Data Pipelines. 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Portoroz, Slovenia, 2020.